

# Automated Semantic Labelling of 3D Vector Models for Scan-to-BIM

Maarten Bassier and Maarten Vergauwen

Dept. of Civil Engineering, TC Construction - Geomatics

KU Leuven - Faculty of Engineering Technology

Ghent, Belgium

Email: maarten.bassier@kuleuven.be

Bjorn Van Genechten

Leica Geosystems

Hexagon

Brussels, Belgium

Email: Bjorn.VanGenechten@leica-geosystems.com

**Abstract**—With the increasing popularity of Building Information modelling (BIM), the demand for accurate as-built models of existing buildings is rising. However, the manual creation of these models is labour intensive and error prone. Therefore, automation of the process is a must. One of the key factors in the automated Scan-to-BIM process is the labelling of the data for further reconstruction. Currently, semantic labelling is still ongoing research.

This paper presents a flexible method to automatically label highly cluttered vector models of existing buildings. In our proposed method, a reasoning framework is used that exploits geometric and contextual information. A major advantage to our approach is that our algorithm can label both cluttered environments and large data sets very efficiently. Unlike other solutions, this allows us to label entire buildings at once. In addition, the implementation of our algorithm and the platform we use allows for flexible data processing, visualisation of the results and improvement of the labelling process. Our work covers the entire labelling phase and allows the user to label data sets with a minimal amount of effort.

**Keywords** - Semantic labelling; Scan-to-BIM; Vector model; Building modelling

## I. INTRODUCTION

The popularity of intelligent three dimensional data models like Building Information Modelling (BIM) is rapidly increasing. Most commonly, these models are created during the designing phase of a structure to support the construction process. Afterwards, stakeholders can employ the BIM model for a wide variety of applications such as facility management, energy performance analysis, project planning, etc [1], [2], [3]. However, the BIM model created during the designing phase often deviates from the actual conditions. Therefore, the need exists for as-built BIM models, where the as-design models are updated to as-built conditions. Experiencing the advantages of

BIM, the industry now looks to implement as-built models for existing buildings. With no prior BIM available, these models have to be created from scratch. In general, as-built models are created from point clouds. Commonly, a terrestrial laser scanner is employed to acquire scans from different locations. The individual data sets are then aligned using cloud-based or target-based registration techniques [4]. Once a complete point cloud is acquired, the data is exported to a modelling software and BIM objects are fitted onto the points. This process is titled

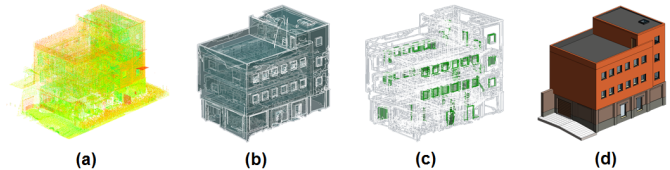


Fig. 1. Overview of intermediate results of the Scan-To-BIM Workflow: From left to right, Point Cloud (a), Vector model (b), labelled Vector model (c) and as-built BIM model (d).

Scan-to-BIM. However, the creation of such a model is labour intensive and time consuming. One of the main problems is the amount of manual labour required in the modelling process. Therefore, the industry would greatly benefit from automation in the Scan-to-BIM process [5], [6], [7]. Our goal is to develop methods to automate this workflow.

## A. Modelling Process

The Scan-to-BIM process can be divided into several phases [8]. Some intermediate results of the different stages are shown in Fig. 1. First, the point cloud is segmented into different clusters. Many solutions have been presented to tackle this problem [9], [10], [11]. Once the clusters are identified, primitives or meshes are fitted onto the data. While complex surfaces provide a more accurate approximation of the as-built conditions, planar surfaces are generally employed to model the structural elements of buildings. Commercial software like Pointfuse [12] already provide fully automated plane reconstruction. In the robotics industry, vector models are often considered as the deliverable of the reconstruction process [13], [14], [15]. However, in the case of Scan-to-BIM, intelligence should be added to the model. This phase is titled "Semantic Labelling". The vector model is processed by reasoning frameworks that provide the individual surfaces with labels such as walls, floors, ceilings, etc. Several approaches have been presented using heuristic or probabilistic techniques [16], [17], [18]. However, most of these solutions only handle small scale data e.g. a single room. Our method provides a reasoning framework that will work on any scale of data, even entire buildings. Considering complete structures allows us to consider some contextual information that is not

available in smaller data sets. Following the semantic labelling, the individual surfaces can be grouped and used as a basis to reconstruct the as-built BIM.

### B. Existing buildings

The focus of this research is on the reconstruction of existing buildings e.g. hospitals, office buildings, schools, houses etc. It is important to know that the data sets of these structures have varying properties. Some of the key factors for the semantic labelling are:

a) *Noise*: The goal of our research is provide an algorithm that works for realistic data. This means that furniture and clutter (such as small objects, persons, etc) will be present in the data sets. This introduces a high degree of confusion in labelling since some furniture can show great resemblance to structural elements. E.g. a built-in closet can be mistaken for a wall or several adjacent tables can be seen as a floor.

b) *Varying zones*: Typical for real structures is the wide variety of zones inside the building. They contain staircases, attics or other unusual spaces. The characteristics of these zones deviate from regular rooms which causes problems in the labelling process.

c) *Type of data set*: Depending on project deliverables and methodology, the type of point cloud data differs. Some data sets only contain a portion of a structure while others consist out of the entire building and the surroundings. Common examples of different types of data sets are *multi-storey* and *single-storey*: The former represents a data set that contains the entire structure (both interior and exterior across all floors). The latter only contains a part of the building, consisting of one floor. The use of single-storey data sets is a common strategy in the Scan-to-BIM industry since treating each floor as a separate project has computational advantages. However, the data characteristics of both types are inherently different. For instance, single-storey data sets do not contain some of the contextual information present in multi-storey data sets. This proves problematic for reasoning algorithms. E.g. a potential ceiling cannot not have a floor above in a single-storey data set. Therefore, filters employing this information are meaningless. Even worse, some filters even have a counterproductive impact on the labelling if they expect information that is not present in the data.

### C. Automation

Traditional, manual modelling relies on the user to interpret the point cloud. In this case, the segmentation of the data, the labelling and the primitive fitting is purely done visually. It is up to the operator to identify the type of object and where to place it.

Automation of the Scan-to-BIM process looks to aid the user in the different phases. Two approaches are currently being developed: *Fully Automated Reconstruction* and *Assisted Manual Reconstruction*. The emphasis of the former is on the creation of an initial proposal of the objects, effectively

removing all user input. Commercial software such as Edge-wise [19] focus on this strategy. On the other hand, the latter strives to provide the user with a set of tools that facilitate the modelling process. Software pursuing this course are FARO PointSense [20] and Leica Cloudworx [21]. While the Fully Automated Reconstruction process is faster, it is limited to generic objects, and thus can only provide an initial solution. The Assisted Manual Reconstruction can aid the operator to a further extent but requires additional labour. We believe a promising solution is to merge the two approaches. For example, automated algorithms could provide an initial proposal during each stage of the Scan-to-BIM process. In addition, the user would be provided with a set of tools to easily update or modify the proposals before continuing. The goal of our research is to automatically provide the user with such an initial BIM model on a flexible adjustment platform.

This article describes our recent work on the automation of the Scan-to-BIM process. More specifically, we address the problem of semantic labelling. Our method provides a reasoning framework that labels elements that frequently appear in typical existing buildings. It identifies floors, ceilings, roofs, walls, windows and doors. The output of our algorithm is a labelled data set that can be easily adjusted by the user for further reconstruction.

The rest of this article is organized as follows. A section of related work is presented in II. Our proposed algorithm is explained in section III. The experimental results are shown in section IV. Finally, in section V, the conclusions and future work are presented.

## II. RELATED WORK

Semantic labelling is being investigated from multiple points of view. A lot of work is performed in the area of computer vision, where labelling is often utilized for object recognition. One of the popular approaches is the use of neural networks such as Conditional Random Fields [22], [23]. By connecting several nodes into a graph, probabilistic reasoning allows likelihood maximisation of the different labels for the nodes. Researchers employing this approach have published promising results for small scale building scenes [5], [16], [17], [24], [25], [26]. However, these reasoning algorithms require extensive learning and are computationally expensive. Other approaches present algorithms that employ geometric and contextual rules [18], [27], [28], [29]. Typical, they encode "features", which represents a characteristic of the candidate object. These features are used to specify a set of rules. E.g. a candidate surface with a large area will have a higher probability of being a structural element. Geometric features found in publications contain normals, dimensions, distance to bounding box, area, height, point density, aspect ratio, etc. Contextual features can consist of associative or non-associative relationships [30]. Common features employed are coplanarity, convexity, proximity, geometric similarity, topology, texture similarity, etc. Most works only employ some of these rules to identify the zone specific elements of the

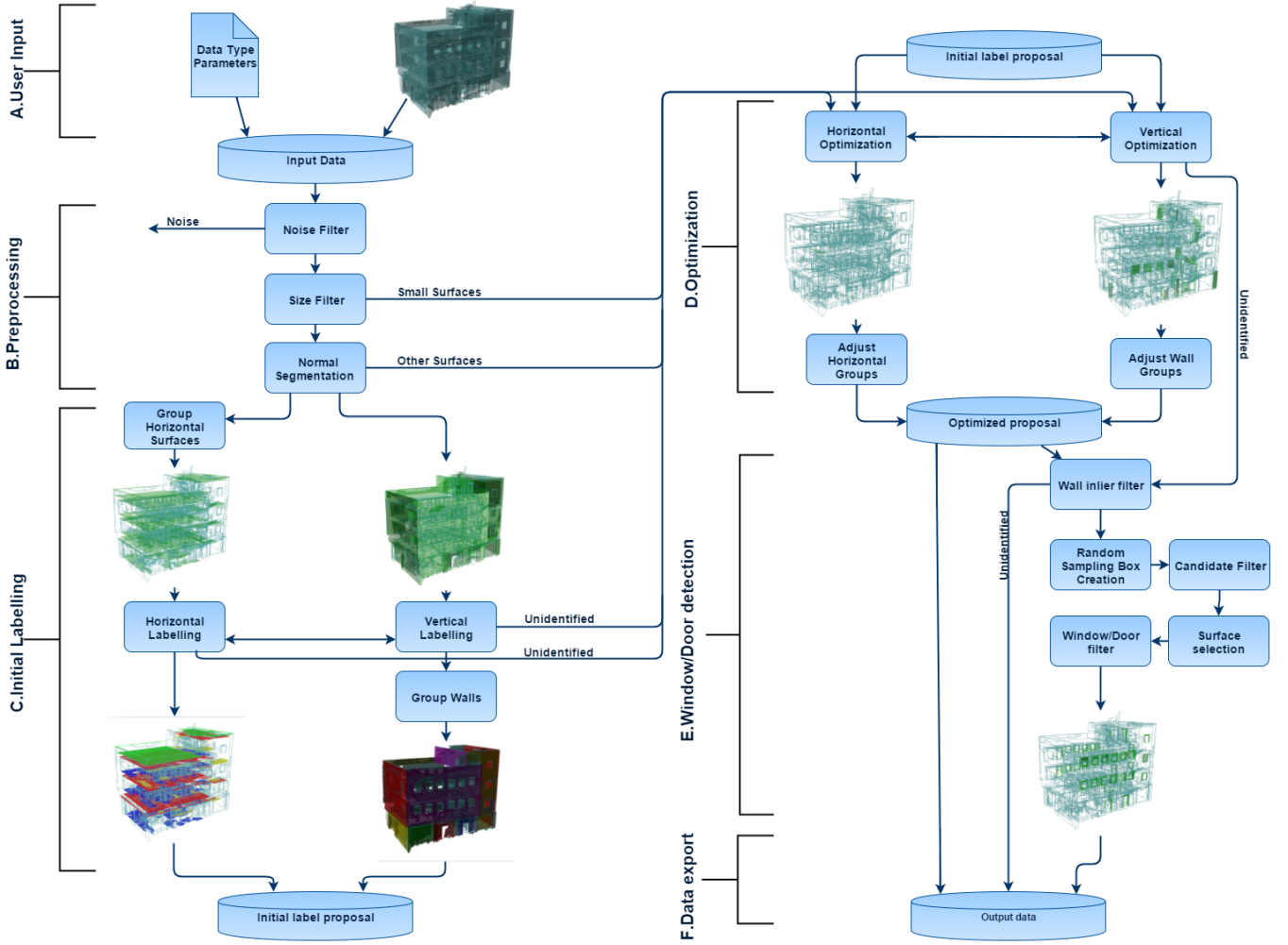


Fig. 2. Workflow diagram of our algorithm. From top left to bottom right the following steps are depicted: User Input (A), Preprocessing (B), Initial labelling (C), Optimization (D), Window/Door detection (E) and Data Export (F).

data set. Our context-based labelling algorithm extrapolates these rules to full scale buildings. Furthermore, different sets of weights for the rules are predefined for varying zones so that the user no longer has to train the algorithm. The input data differs between varying approaches. Some researchers directly segment the point cloud [13] while others prefer to work with primitives such as surfaces. Although working directly on the point clouds can be more accurate, it also introduces a higher computational cost and uncertainty in the process. In our work, we use primitives as a basis because of the computational advantages. Some research has been performed on the automated reconstruction of rooms based on labelling [31], [32]. The graph based techniques in these works provide promising labelling results, but requires a watertight mesh. However, the creation of such a mesh in a highly cluttered and occluded environment is challenging. Another point of view is the use of prior knowledge. [33], [34] employ contextual information on existing plans to extract building

elements. However, these plans often lack consistency making them hard to interpret. Semantic labelling has also been a major topic in outdoor scenes. Facade and window opening detection algorithms have been successfully used on mobile mapping data [35]. In aerial applications, building extraction has seen major breakthroughs in the past several years [29], [36], [37], [38], [39]. However, these algorithms only employ exterior information. Our algorithm exploits both indoor and outdoor data to find accurate walls, roofs and openings.

### III. PROPOSED WORKFLOW

In this paper we consider flexible labelling of planar surfaces for the creation of BIM models for existing buildings. Our approach consists of a weighted reasoning framework that employs geometric and contextual rules to label the data set.

Our algorithm is implemented in Grasshopper, an open source Rhinoceros plug-in which is a platform for visual scripting of object based program languages. The algorithm

pipeline shown in Fig. 2 consists of six steps. First, before the actual processing, the user chooses a set of parameters in respect to the data. Second, preprocessing of the data is performed for noise reduction and segmentation. Third, the initial labelling is computed. Fourth, an optimization step is performed to enhance the labelling. The fifth step allows the operator to automatically find window and door surfaces. Finally, the sixth step exports the labelled surfaces to Rhinoceros, where the data can be updated by the user.

As an input, our algorithm accepts any set of flattened meshes. This geometry can be derived from multiple reconstruction software. The example input meshes used in this paper are produced by Pointfuse [12]. This software calculates flat triangular meshes to represent the planes in the point cloud.

#### A. User Input

The platform we currently employ is Rhinoceros. The input meshes are loaded into the software and visualized. Before any calculations are performed, the user has the opportunity to evaluate the data and choose appropriate settings in the Grasshopper interface: First, the user defines which part of the data should be processed. Second, the user indicates which type of data the algorithm should be expecting (E.g. multi-storey). Varying types of data have different preset parameters for the contextual and geometric filters. These parameters are previously trained by example data.

An advantage of our approach is that our algorithm allows flexible zone and parameter set selection. While an entire project can be loaded into the algorithm at once, it is possible to select only a part of the data at a time and feed it to the labelling algorithm with a specific set of parameters. This way, the user maintains control of the labelling process. Also, the user can easily create and train new sets of parameters for other types of data.

#### B. Preprocessing

At the beginning of the preprocessing step, the data is imported from Rhinoceros into the Grasshopper plug-in. There, planar surfaces are extracted from the meshes based on the exterior boundaries of the meshes. The advantage of the planar surface representation is that the same geometric data can be handled more efficiently. Following, two dimension filters are applied: One for noise reduction and another to segment the large from the small surfaces. By splitting the data set by size, we allow for coarse to fine labelling. The noise dimension threshold is set to 0.7m for the major axis. The segmentation filter threshold considers surfaces to be large if both axes are larger than 1m or the major axis exceeds 2.5m. Next, the large surfaces are divided by their normal with an angular threshold of 15° into *Horizontal Surfaces* and *Vertical Surfaces*. Surfaces with a normal between 15° and 85° are considered as *Other Surfaces*. These surfaces are not labelled because of their unpredictable characteristics and low occurrence (less than 1% of data set).

#### C. Initial labelling

In this step, each surface in  $S = \{s_1, s_2, \dots, s_n\}$  is assigned one of the following labels: The horizontal surfaces are divided into *Floors*, *Ceilings*, *Roofs*, *Furniture* and *Unidentified Horizontal*. The vertical surfaces are labelled *Wall* or *Unidentified Vertical*. The labelling is performed by testing each surface with a set of contextual and geometric filters  $F_l = \{f_{l,1}, f_{l,2}, \dots, f_{l,n}\}$ . The features employed by the filters are shown in table I. For instance, the *Area*, *Proximity* and *Extrusion Collision* are evaluated in equations (1), (2), (3) respectively.

$$f_{Area}(s, S) := A_s - \bar{A}_S \leq t_{Area} \quad (1)$$

$$f_{Prox}(s, S) := \|B_s - B_S\| \leq t_{Prox} \quad (2)$$

$$f_{Extr}(s, S) := \#\{f_{Extr}(s) \cap S_{large}\} \leq t_{Extr} \quad (3)$$

Each filter yields a discrete value 0 or 1 for each surface (4) based on a threshold  $t_i$ . These filter specific thresholds define whether a relation is valid or not. E.g. the euclidean distance of 0.5m between the borders of two surfaces is set as a threshold to define a neighbourhood relationship.

$$f_{l,i}(s, S) = \begin{cases} 1, & f_i(s, S) \leq t_i \\ 0, & f_i(s, S) > t_i \end{cases} \quad (4)$$

A weighted voting function (5) for each label is implemented to distribute the importance of every filter. The filter weights  $W_l = \{w_{l,1}, w_{l,2}, \dots, w_{l,n}\}$  are preset for every label and are dependant on the type of data. The sum of these weights for a specific label amounts to 1 (5). A minimum label value  $V_l$  of 0.7 in the voting function is required for a surface to be labelled. Once the value exceeds this threshold, the label with the highest value is assigned to the surface. Surfaces with a lower value are labelled respectively *Unidentified Horizontal* or *Unidentified Vertical*.

TABLE I  
GEOMETRIC AND CONTEXTUAL FEATURES.

Labels	Geometric features	Contextual features
Floor Ceiling Roof Furniture	Area Dimensions Boundary Normal	Wall proximity Wall height similarity Horizontal proximity Extrusion collision Room edge proximity
Wall	Area Dimensions Boundary Normal	Ceiling proximity Floor proximity Coplanarity Perpendicularity Wall proximity Room edge proximity
Window Door	Diagonal length Diagonal angle Dimensions	Floor proximity Wall inlier
<b>Threshold</b>	<b>0.70</b>	<b>0.70</b>

TABLE II  
INDIVIDUAL LABEL PERFORMANCE OF "SCHOOL" DATA SET.

	Noise	Floor	Ceiling	Roof	Wall	Window	Door	Unidentified	Overall
Total Surfaces	1129	21	19	7	134	305	99	1464	<b>3178</b>
Total Elements	-	7	10	4	47	28	12	-	<b>108</b>
Surface Precision [%]	-	100	83.3	85.7	91.1	94.7	86.8	-	<b>90.3</b>
Element Precision [%]	-	100	72.7	85.7	87.5	88.2	83.3	-	<b>86.2</b>
Surface Recall [%]	-	85.7	90.5	100	91.8	91.2	38.4	-	<b>82.9</b>
Element Recall [%]	-	85.7	80	100	100	96.4	41.6	-	<b>84.0</b>
Computation time [s]	2.8	3.5			13.9	17.6		-	<b>37.8</b>

$$\sum_{f_{l,i} \in F_l} f_{l,i}(s, S) w_{l,i} \geq V_l \quad \text{where} \quad \sum_{w_{l,i} \in W_l} w_{l,i} = 1 \quad (5)$$

In this step, our algorithm focusses on precision instead of recall to provide a reliable basis for the optimization step.

#### D. Optimization

The emphasis of the optimization step is on the maximisation of recall values for the different labels. Once the larger objects have been identified in the initial labelling, the remaining surfaces are evaluated and labelled. During this step, several features are evaluated based on the initial labelling. First, room information is used to find additional floors, ceilings and walls. The rooms are computed utilizing the initial labels of the floors and the ceilings. The ceiling surfaces are the principal component of the room boundaries since they are often less occluded than the floor surfaces. Spatial filtering is applied on the remaining surfaces to label additional surfaces based on room boundary proximity. Second, wall height similarity in combination with wall proximity is used to label additional walls, ceilings and floors. The result of this step is a labelled data set of the structural elements.

#### E. Window/Door detection

After the optimization step, our algorithm employs the wall information as a basis to find windows and doors. First, a spatial filter is applied on the remaining data to isolate the surfaces that are located inside the walls. For single-faced walls, a local search area is defined to locate nearby surfaces. The data extracted from both filters serve as the candidate surfaces for the window and door detection step.

Second, conditional random sampling is applied to locate candidate bounding boxes per wall. As minimal sample, one vertical and one horizontal surface is iteratively selected at random from each wall to create a bounding box. The process is conditioned so that only boxes are created that have similar dimensions as the initial samples. After the candidates are created, a reasoning framework is applied using length of diagonal, angle of diagonal, similar centroid occurrences and surface inliers to filter the bounding boxes. Surfaces having more than 30% overlap with a filtered bounding box are

withheld and grouped. Finally, windows are separated from doors based on dimensions and floor information.

#### F. Data export

After the labelling is completed, the data is exported to their respective layers in the Rhinoceros software. There, the user can easily improve the data to a fully labelled data set. Once the adjustments are completed, the data is re-imported into the Grasshopper environment for future reconstruction.

### IV. EXPERIMENTAL RESULTS

We tested our algorithm using data from several existing structures. For research purposes, only multi-storey data sets were selected due to their more complex and complete nature. To provide a realistic data set, the point clouds were not cleaned after registration. Also, the generated meshes were not altered in any way. Two data sets are used to evaluate our labelling algorithm: A school building and a house. Both data sets have slightly different features but can be considered to be the same data type. We tested our algorithm on both recall and precision performances for each label.

a) *School building*: The first test case is a school building on the technology campus in Ghent. The building has four floors and houses a laboratory, two classrooms, a staircase and a maintenance room. 40 scans were acquired for a total of 400 million points. The Pointfuse software was able to reconstruct 3178 surfaces with an average standard deviation of 0.001m. The entire data set was processed at once. We compared our results against a manually labelled data set which was used as a benchmark. Table II and Fig. 3 (a to i) depict the results of the process. Noticeable is the amount of noise present in the data set (Fig. 3, g). No less than 35.5% of the data is too small to participate in the structural element detection. On average, 82.9% of the surface labels were found, resulting in a 84% element detection. With average precisions of 90.3% and 86.2%, the selection is very accurate. Fig. 3, h shows that the only floors and ceilings left unlabelled are part of the staircase. This is to be expected since the surfaces in the staircase have deviating features compared to the floor and ceiling features in more generic rooms. The percentage of doors found is far less than other elements. However, looking at Fig. 3 (e and f), it is revealed that some of the doors were labelled as windows and



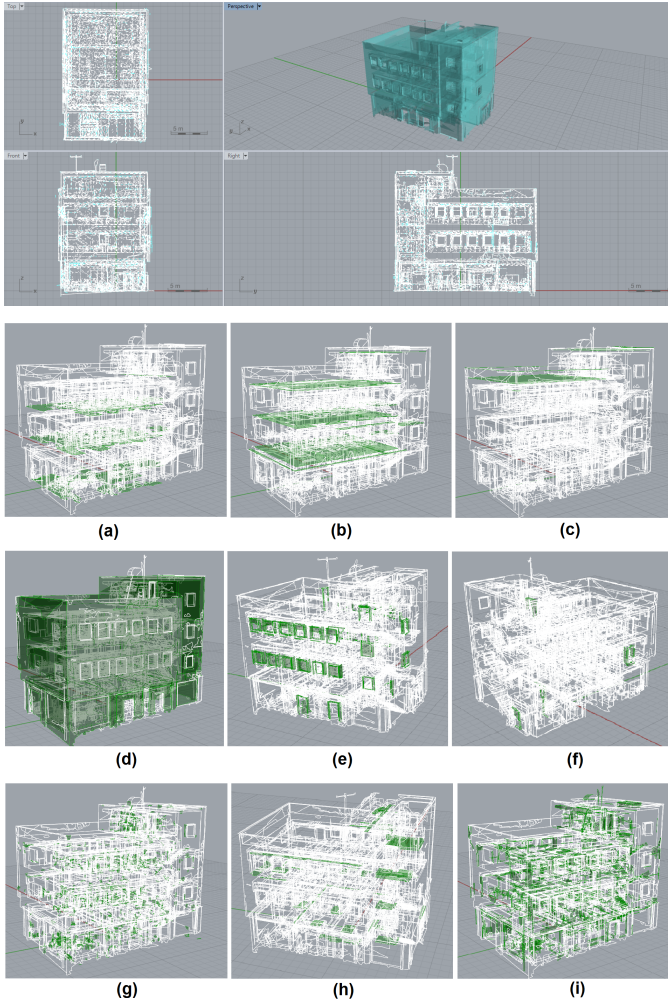


Fig. 3. Overview of the results of the "School" data set: the top four images contain views of the data set. The bottom figures show the results for the labelling process: The floor surfaces (a), the ceiling surfaces (b), the roof surfaces (c), the wall surfaces (d), the windows surfaces (e), the door surfaces (f), the noise surfaces (g), the unidentified large surfaces (h) and the unidentified small surfaces (i).

vice versa. This error can easily be adjusted by the user since the elements are found nonetheless. The unidentified surfaces (Fig. 3, i) are mainly a mixture of small parts of elements and noise.

The computational effort of our algorithm is relatively low. A total of 37.8s was required to label the data sets and calculate the previews of the entire building. The bottom row in Table II shows that the window and door detection algorithm is the slowest step. This is caused by the large number of iterations in the random sampling to find sufficient support for the bounding box candidates.

*b) House:* The second data set is a terraced house. The building has four floors and houses several bedrooms, a living room, a kitchen, two bathrooms, a basement and some other small rooms. 35 scans were acquired for a total of 350 million points. The scene is highly cluttered and the structural

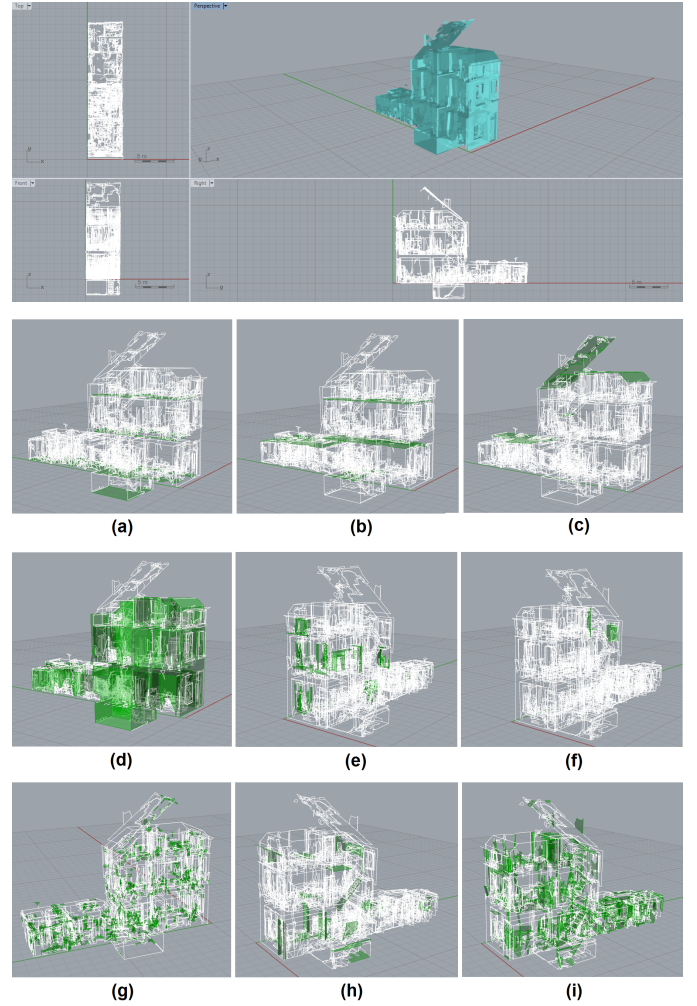


Fig. 4. Overview of the results of the "House" data set: the top four images contain views of the data set. The bottom figures show the results for the labelling process: The floor surfaces(a), the ceiling surfaces(b), the roof surfaces(c), the wall surfaces(d), the windows surfaces(e), the door surfaces(f), the noise surfaces(g), the unidentified large surfaces(h) and the unidentified small surfaces(i).

elements are partially occluded. A total of 1989 surfaces were reconstructed with an average standard deviation of 0.001m. Fig. 4 (a to i) and table III give an overview of the labelling results.

Table III reveals slightly poorer results for the second data set. Average recall values of 74.5% and 76.0% were computed for the surfaces and the elements. The largest difference is located in the window and door detection phase. This can be explained by the clutter and occlusion present in the scene. When looking at the structural component detection, both data sets reveal similar results (respectively 92.4% and 87.8% for data set 1 and 2 for the element detection). Also, the precision values show similar results (86.3% for the surfaces and 80.2% for the elements). Analysing the computational effort, a total of 20.0s was timed for the entire process. From table II and III can be derived that an increase of 37.4% in surfaces results in

TABLE III  
INDIVIDUAL LABEL PERFORMANCE OF "HOUSE" DATA SET.

	Noise	Floor	Ceiling	Roof	Wall	Window	Door	Unidentified	Overall
Total Surfaces	999	14	17	9	98	95	84	673	<b>1989</b>
Total Elements	-	5	8	3	50	7	12	-	<b>85</b>
Surface Precision [%]	-	100	93.3	100	90.6	70.2	63.6	-	<b>86.3</b>
Element Precision [%]	-	100	87.5	100	88	55.6	50.0	-	<b>79.7</b>
Surface Recall [%]	-	85.7	88.2	100	76.5	83.6	13.1	-	<b>74.5</b>
Element Recall [%]	-	80	77.8	100	100	81.3	16.7	-	<b>76.0</b>
Computation time [s]	1.6	1.7			6.4	10.3		-	<b>20.0</b>

an increase of 46.0% in computation time. Also it is revealed that our solution primarily focusses on precision instead of recall (86.3% opposed to 74.5% for both data sets). While this reduces overall errors, it is not optimal for our flexible adjustment approach. We prefer recall over precision since it is easier to delete false positives than to find false negatives.

## V. CONCLUSION & FUTURE WORK

In this paper we explained and demonstrated a flexible automated labelling framework for existing buildings for the Scan-to-BIM pipeline. In our proposed method, a reasoning framework is employed that exploits geometric and contextual information. A major advantage to our approach is that our algorithm can label both cluttered environments and large data sets very efficiently. Unlike other solutions, this allows us to label entire buildings at once. Furthermore, the implementation of our algorithm and the platform we use allows for flexible data processing, visualisation of the results and improvement of the labelling process. Our work covers the entire labelling phase and allows the user to complete the labelling with a minimal amount of effort.

Overall, the experimental data showed that our algorithm is able to label the general structural elements in common areas with high precision and recall values. However, in more complex areas such as staircases, the features of the surfaces differ from the features in common areas. This proves problematic for the surface labelling. Also, elements that consist of small parts are harder to detect because their surface features resemble features of noise surfaces. For these types of data our algorithm will underperform. Currently our approach relies on user input for zone and parameter selection. Work is being performed on the automation of this input, further reducing the users effort.

Currently, the parameters in the algorithm are focussed on the maximisation of precision instead of recall. However, our approach would benefit from recall maximisation because of the flexible adjustment platform. It is easier for the user to remove false positives than to add false negatives. In order to maximise recall, the contribution of each parameter should be known. Therefore, further work is focussed on key parameter

identification and recall optimization using machine learning techniques.

## REFERENCES

- [1] E. M. Wetzel and W. Y. Thabet, "The use of a BIM-based framework to support safe facility management processes," *Automation in Construction*, vol. 60, pp. 12–24, 2015. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0926580515001971>
- [2] S. Kalavagunta, R. Jayaram, and S. Naganathan, "Energy Performance Analysis of Buildings in BIM Workflow," in *3rd ASEAN - Australian Engineering Congress 2015 On Innovative Technologies in Sustainability Development and Renewable Energy*, no. March, 2015.
- [3] B. Becerik-Gerber, F. Jazizadeh, N. Li, and G. Calis, "Application Areas and Data Requirements for BIM-Enabled Facilities Management," *Journal of Construction Engineering and Management*, vol. 138, no. 3, pp. 431–442, 2012. [Online]. Available: [http://ascelibrary.org/doi/10.1061/\(ASCE\)CO.1943-7862.0000433](http://ascelibrary.org/doi/10.1061/(ASCE)CO.1943-7862.0000433)
- [4] M. Bassier, M. Yousefzadeh, and B. V. Genechten, "Evaluation of data acquisition techniques and workflows for Scan to BIM," in *Geo Business*, 2015.
- [5] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques," *Automation in Construction*, vol. 19, no. 7, pp. 829–843, nov 2010. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0926580510000907>
- [6] I. Brilakis, M. Lourakis, R. Sacks, S. Savarese, S. Christodoulou, J. Teizer, and A. Makhmalbaf, "Advanced Engineering Informatics Toward automated generation of parametric BIMs based on hybrid video and laser scanning data," *Advanced Engineering Informatics*, vol. 24, no. 4, pp. 456–465, 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.aei.2010.06.006>
- [7] N. Hichri, C. Stefani, L. D. Luca, P. Veron, and G. Hamon, "FROM POINT CLOUD TO BIM : A SURVEY OF EXISTING APPROACHES," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5/W2, no. September, pp. 2–6, 2013.
- [8] V. Ptrucean, I. Armeni, M. Nahangi, J. Yeung, I. Brilakis, and C. Haas, "State of research in automatic as-built modelling," *Advanced Engineering Informatics*, vol. 29, pp. 162–171, 2015. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1474034615000026>
- [9] S. Barnea and S. Filin, "Segmentation of terrestrial laser scanning data using geometry and image information," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 76, pp. 33–48, feb 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0924271612000809>
- [10] H. Woo, E. Kang, S. Wang, and K. H. Lee, "A new segmentation method for point cloud data," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 2, pp. 167–178, 2002.
- [11] T. Rabbani, F. a. van den Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraint," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences - Commission V Symposium 'Image Engineering and Vision Metrology'*, vol. 36, no. 5, pp. 248–253, 2006. [Online]. Available: [http://www.isprs.org/proceedings/XXXVI/part5/paper/RABB\\_639.pdf](http://www.isprs.org/proceedings/XXXVI/part5/paper/RABB_639.pdf)
- [12] L. Arithmetica, "pointfuse," 2015. [Online]. Available: <http://pointfuse.com/>

- [13] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 286–304, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.isprsjprs.2015.01.016>
- [14] A. J. B. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient Organized Point Cloud Segmentation with Connected Components."
- [15] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Transactions on Graphics*, vol. 32, no. 4, p. 1, jul 2013. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2461912.2461919>
- [16] X. Xiong, A. Adan, B. Akinci, and D. Huber, "Automatic creation of semantically rich 3D building models from laser scanner data," *Automation in Construction*, vol. 31, pp. 325–337, may 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0926580512001732>
- [17] R. B. Rusu, Z. C. Marton, N. Blodow, A. Holzbach, and M. Beetz, "Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, no. November, pp. 3601–3608, 2009.
- [18] S. Pu and G. Vosselman, "Knowledge based reconstruction of building models from terrestrial laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 6, pp. 575–584, nov 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0924271609000501>
- [19] ClearEdge, "Case Study : University Science Building As-Built Revit Model EdgeWise Building Delivers a 55 % Workflow Improvement for Scan-to-Revit Modeling," p. 8210, 2013. [Online]. Available: <http://www.clearedge3d.com/>
- [20] FARO PointSense, "Processing of 3D laser scanning data," 2015. [Online]. Available: <http://www.faro-3d-software.com/>
- [21] L. Geosystems, "Leica Cloudworxx 2.0: Revit Point cloud plug-in software," pp. 2–3, 2014. [Online]. Available: <http://www.leica-geosystems.com/>
- [22] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, jan 2009. [Online]. Available: <http://www.jstor.org/stable/478142?origin=crossref>
- [23] A. Criminisi and J. Shotton, *Decision forests for computer vision and medical image analysis*. Springer London Heidelberg New York Dordrecht, 2013.
- [24] X. Xiong and D. Huber, "Using Context to Create Semantic 3D Models of Indoor Environments," *Proceedings of the British Machine Vision Conference 2010*, pp. 45.1–45.11, 2010. [Online]. Available: <http://www.bmva.org/bmvc/2010/conference/paper45/index.html>
- [25] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas, "Acquiring 3D indoor environments with variability and repetition," *ACM Transactions on Graphics*, vol. 31, no. 6, p. 1, nov 2012. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2366145.2366157>
- [26] L. Tamas and L. Cosmin Goron, "3D semantic interpretation for robot perception inside office environments," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 76–87, jun 2014. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0952197614000566>
- [27] C. Langenhan, M. Weber, M. Liwicki, F. Petzold, and A. Dengel, "Graph-based retrieval of building information models for supporting the early design stages," *Advanced Engineering Informatics*, vol. 27, no. 4, pp. 413–426, 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1474034613000499>
- [28] C. Wang, Y. K. Cho, and C. Kim, "Automatic BIM component extraction from point clouds of existing buildings for sustainability applications," *Automation in Construction*, vol. 56, pp. 1–13, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0926580515000734>
- [29] H. Lin, J. Gao, Y. Zhou, G. Lu, M. Ye, C. Zhang, L. Liu, and R. Yang, "Semantic decomposition and reconstruction of residential scenes from LiDAR data," *ACM Transactions on Graphics*, vol. 32, no. 4, p. 1, jul 2013. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2461912.2461969>
- [30] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic Labeling of 3D Point Clouds for Indoor Scenes," in *R:SS workshop on RGB-D cameras*, 2013, pp. 1–9.
- [31] E. Turner, "Floor Plan Generation and Room Labeling of Indoor Environments from Laser Range Data Floor Plans," 2014.
- [32] S. Ochmann, R. Vock, R. Wessel, M. Tamke, and R. Klein, "Automatic Generation of Structural Building Descriptions from 3D Point Cloud Scans," in *GRAPP 2014 - International Conference on Computer Graphics Theory and Applications*, 2014.
- [33] H. B. Hmida, C. Cruz, F. Boochs, C. Nicolle, and U. D. Bourgoigne, "From Unstructured 3D Point Clouds to Structured Knowledge - A Semantics Approach," *Advances in Theories and Mathematical Models*, 2011.
- [34] H. Son, F. Bosché, and C. Kim, "As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 172–183, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S147403461500021X>
- [35] R. Wang, J. Bach, and F. P. Ferrie, "Window detection from mobile LiDAR data," *2011 IEEE Workshop on Applications of Computer Vision, WACV 2011*, pp. 58–65, 2011.
- [36] A. M. Ramiya, R. R. Nidamanuri, and R. Krishnan, "Object-oriented semantic labelling of spectralspatial LiDAR point cloud for urban land cover classification and buildings detection," *Geocarto International*, vol. 6049, no. July, pp. 1–19, 2015. [Online]. Available: <http://www.tandfonline.com/doi/full/10.1080/10106049.2015.1034195>
- [37] I. Jazayeri, A. Rajabifard, and M. Kalantari, "A geometric and semantic evaluation of 3D data sourcing methods for land and property information," *Land Use Policy*, vol. 36, pp. 219–230, jan 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0264837713001543>
- [38] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of lidar data and building object detection in urban areas," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 152–165, jan 2014. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0924271613002359>
- [39] R. Richter, M. Behrens, and J. Döllner, "Object class segmentation of massive 3D point clouds of urban areas using point cloud topology," *International Journal of Remote Sensing*, vol. 34, no. 23, pp. 8408–8424, dec 2013. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/01431161.2013.838710>